

你知道关于移动端开发的技巧有哪些？

META 相关

1. 添加到主屏后的标题（IOS）

```
<meta name="apple-mobile-web-app-title" content="标题">
```

2. 启用 WebApp 全屏模式（IOS）

当网站添加到主屏幕后再点击进行启动时，可隐藏地址栏（从浏览器跳转或输入链接进入并没有此效果）

```
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-touch-fullscreen" content="yes" />
```

3. 百度禁止转码

通过百度手机打开网页时，百度可能会对你的网页进行转码，往你页面贴上它的广告，非常之恶心。不过我们可以通过这个 meta 标签来禁止它：

```
<meta http-equiv="Cache-Control" content="no-siteapp" />
```

4. 设置状态栏的背景颜色（IOS）

设置状态栏的背景颜色，只有在 "apple-mobile-web-app-capable" content="yes" 时生效

```
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />
```

content 参数：

- default：状态栏背景是白色。
- black：状态栏背景是黑色。
- black-translucent：状态栏背景是半透明。如果设置为 default 或 black，网页内容从状态栏底部开始。如果设置为 black-translucent，网页内容充满整个屏幕，顶部会被状态栏遮挡。

5. 移动端手机号码识别（IOS）

在 iOS Safari（其他浏览器和 Android 均不会）上会对那些看起来像是电话号码的数字处理为电话链接，比如：

- 7 位数字，形如：1234567
- 带括号及加号的数字，形如：(+86)123456789
- 双连接线的数字，形如：00-00-00111
- 11 位数字，形如：13800138000

可能还有其他类型的数字也会被识别。我们可以通过如下的 meta 来关闭电话号码的自动识别：

```
<meta name="format-detection" content="telephone=no" />
```

开启电话功能

```
<a href="tel:123456">123456</a>
```

开启短信功能：

```
<a href="sms:123456">123456</a>
```

6. 移动端邮箱识别 (Android)

与电话号码的识别一样，在安卓上会对符合邮箱格式的字符串进行识别，我们可以通过如下的 `meta` 来管别邮箱的自动识别：

```
<meta content="email=no" name="format-detection" />
```

同样地，我们也可以通过标签属性来开启长按邮箱地址弹出邮件发送的功能：

```
<a mailto:dooyoe@gmail.com">dooyoe@gmail.com</a>
```

7. 添加智能 App 广告条 Smart App Banner (iOS 6+ Safari)

```
<meta name="apple-itunes-app" content="app-id=myAppStoreID, affiliate-  
data=myAffiliateData, app-argument=myURL">
```

8. iOS Web app 启动动画

由于 iPad 的启动画面是不包括状态栏区域的。所以启动图片需要减去状态栏区域所对应的方向上的 20px 大小，相应地在 retina 设备上要减去 40px 的大小

```
<!-- iPhone -->  
<link href="apple-touch-startup-image-320x460.png" media="(device-width: 320px)"  
rel="apple-touch-startup-image">  
<!-- iPhone (Retina) -->  
<link href="apple-touch-startup-image-640x960.png" media="(device-width: 320px)  
and (-webkit-device-pixel-ratio: 2)" rel="apple-touch-startup-image">  
<!-- iPad (portrait) -->  
<link href="apple-touch-startup-image-768x1004.png" media="(device-width: 768px)  
and (orientation: portrait)" rel="apple-touch-startup-image">  
<!-- iPad (landscape) -->  
<link href="apple-touch-startup-image-748x1024.png" media="(device-width: 768px)  
and (orientation: landscape)" rel="apple-touch-startup-image">  
<!-- iPad (Retina, portrait) -->  
<link href="apple-touch-startup-image-1536x2008.png" media="(device-width:  
1536px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2)"  
rel="apple-touch-startup-image">  
<!-- iPad (Retina, landscape) -->  
<link href="apple-touch-startup-image-2048x1496.png" media="(device-width:  
1536px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2)"  
rel="apple-touch-startup-image">
```

(landscape: 横屏 | portrait: 竖屏)

9. 添加到主屏后的 APP 图标

指定 web app 添加到主屏后的图标路径，有两种略微不同的方式：

```
<!-- 设计原图 -->
<link href="short_cut_114x114.png" rel="apple-touch-icon-precomposed">
<!-- 添加高光效果 -->
<link href="short_cut_114x114.png" rel="apple-touch-icon">
```

- **apple-touch-icon**: 在 IOS6 及以下的版本会自动为图标添加一层高光效果（IOS7 开始已使用扁平化的设计风格）
- **apple-touch-icon-precomposed**: 使用“设计原图图标”

效果：



图标尺寸：

可通过指定 **size** 属性来为不同的设备提供不同的图标（但通常来说，我们只需提供一个 114 x 114 pixels 大小的图标即可）

官方说明如下

Create different sizes of your app icon for different devices. If you're creating a universal app, you need to supply app icons in all four sizes.

For iPhone and iPod touch both of these sizes are required:

57 x 57 pixels

114 x 114 pixels (high resolution)

For iPad, both of these sizes are required:

72 x 72 pixels

144 x 144 (high resolution)

10. 优先使用最新版本 IE 和 Chrome

```
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
```

11.viewport 模板

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="utf-8">
<meta content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-
scalable=no" name="viewport">
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
<meta content="telephone=no" name="format-detection">
<meta content="email=no" name="format-detection">
<title>标题</title>
<link rel="stylesheet" href="index.css">
</head>
<body>
这里开始内容
</body>
</html>
```

常见问题

1、移动端如何定义字体 font-family

三大手机系统的字体：

ios 系统

- 默认中文字体是 Heiti SC
- 默认英文字体是 Helvetica
- 默认数字字体是 HelveticaNeue
- 无微软雅黑字体

android 系统

- 默认中文字体是 Droidsansfallback
- 默认英文和数字字体是 Droid Sans
- 无微软雅黑字体

winphone 系统

- 默认中文字体是 Dengxian(方正等线体)
- 默认英文和数字字体是 Segoe
- 无微软雅黑字体

各个手机系统有自己的默认字体，且都不支持微软雅黑
如无特殊需求，手机端无需定义中文字体，使用系统默认
英文字体和数字字体可使用 Helvetica，三种系统都支持

* 移动端定义字体的代码 */

```
body{font-family:Helvetica;}
```

2、移动端字体单位 font-size 选择 px 还是 rem

对于只需要适配手机设备，使用 px 即可

对于需要适配各种移动设备，使用 rem，例如只需要适配 iPhone 和 iPad 等分辨率差别比较挺大的设备

rem 配置参考:

```
html {font-size:10px}
@media screen and (min-width:480px) and (max-width:639px) {
  html {
    font-size: 15px
  }
}
@media screen and (min-width:640px) and (max-width:719px) {
  html {
    font-size: 20px
  }
}
@media screen and (min-width:720px) and (max-width:749px) {
  html {
    font-size: 22.5px
  }
}
@media screen and (min-width:750px) and (max-width:799px) {
  html {
    font-size: 23.5px
  }
}
@media screen and (min-width:800px) and (max-width:959px) {
  html {
    font-size: 25px
  }
}
@media screen and (min-width:960px) and (max-width:1079px) {
  html {
    font-size: 30px
  }
}
@media screen and (min-width:1080px) {
  html {
    font-size: 32px
  }
}
```

3、移动端 touch 事件(区分 webkit 和 winphone)

当用户手指放在移动设备在屏幕上滑动会触发的 touch 事件

以下支持 webkit

- touchstart——当手指触碰屏幕时候发生。不管当前有多少只手指

- touchmove——当手指在屏幕上滑动时连续触发。通常我们再滑屏页面，会调用 event 的 preventDefault() 可以阻止默认情况的发生：阻止页面滚动
- touchend——当手指离开屏幕时触发
- touchcancel——系统停止跟踪触摸时候会触发。例如在触摸过程中突然页面 alert() 一个提示框，此时会触发该事件，这个事件比较少用

以下支持 winphone 8

- MSPointerDown——当手指触碰屏幕时候发生。不管当前有多少只手指
- MSPointerMove——当手指在屏幕上滑动时连续触发。通常我们再滑屏页面，会调用 css 的 html{-ms-touch-action: none;} 可以阻止默认情况的发生：阻止页面滚动
- MSPointerUp——当手指离开屏幕时触发

4、移动端 click 屏幕产生 200-300 ms 的延迟响应

移动设备上的 web 网页是有 300ms 延迟的，玩玩会造成按钮点击延迟甚至是点击失效。

以下是历史原因：

2007 年苹果发布首款 iPhone 上 iOS 系统搭载的 Safari 为了将适用于 PC 端上大屏幕的网页能比较好的展示在手机端上，使用了双击缩放(double tap to zoom)的方案，比如你在手机上用浏览器打开一个 PC 上的网页，你可能在看到页面内容虽然可以撑满整个屏幕，但是字体、图片都很小看不清，此时可以快速双击屏幕上的某一部分，你就能看清该部分放大后的内容，再次双击后能回到原始状态。

双击缩放是指用手指在屏幕上快速点击两次，iOS 自带的 Safari 浏览器会将网页缩放至原始比例。

原因就出在浏览器需要如何判断快速点击上，当用户在屏幕上单击某一个元素时候，例如跳转链接 ``，此处浏览器会先捕获该次单击，但浏览器不能决定用户是单纯要点击链接还是要双击该部分区域进行缩放操作，所以，捕获第一次单击后，浏览器会先 Hold 一段时间 t ，如果在 t 时间区间里用户未进行下一次点击，则浏览器会做单击跳转链接的处理，如果 t 时间里用户进行了第二次单击操作，则浏览器会禁止跳转，转而进行对该部分区域页面的缩放操作。那么这个时间区间 t 有多少呢？在 iOS Safari 下，大概为 300 毫秒。这就是延迟的由来。造成的后果用户纯粹单击页面，页面需要过一段时间才响应，给用户慢体验感觉，对于 web 开发者来说是，页面 js 捕获 click 事件的回调函数处理，需要 300ms 后才生效，也就间接导致影响其他业务逻辑的处理。

解决方案：

- [fastclick](#) 可以解决在手机上点击事件的 300ms 延迟
- zepto 的 touch 模块，tap 事件也是为了解决在 click 的延迟问题

触摸事件的响应顺序

- 1、ontouchstart
- 2、ontouchmove
- 3、ontouchend
- 4、onclick

解决 300ms 延迟的问题，也可以通过绑定 ontouchstart 事件，加快对事件的响应

5、什么是 Retina 显示屏，带来了什么问题

retina: 一种具备超高像素密度的液晶屏，同样大小的屏幕上显示的像素点由 1 个变为多个，如在同样带下的屏幕上，苹果设备的 retina 显示屏中，像素点 1 个变为 4 个。在高清显示屏中的位图被放大，图片会变得模糊，因此移动端的视觉稿通常会设计为传统 PC 的 2 倍。

那么，前端的应对方案是：

设计稿切出来的图片长宽保证为偶数，并使用 **background-size** 把图片缩小为原来的 1/2

//例如图片宽高为：200px*200px，那么写法如下

```
.css{width:100px;height:100px;background-size:100px 100px;}
```

其它元素的取值为原来的 1/2，例如视觉稿 40px 的字体，使用样式的写法为 20px

```
.css{font-size:20px}
```

6、ios 系统中元素被触摸时产生的半透明灰色遮罩怎么去掉

ios 用户点击一个链接，会出现一个半透明灰色遮罩，如果想要禁用，可设置 **-webkit-tap-highlight-color** 的 alpha 值为 0，也就是属性值的最后一位设置为 0 就可以去除半透明灰色遮罩

```
a,button,input,textarea{-webkit-tap-highlight-color: rgba(0,0,0,0)}
```

7、部分 android 系统中元素被点击时产生的边框怎么去掉

android 用户点击一个链接，会出现一个边框或者半透明灰色遮罩，不同生产商定义出来效果不一样，可设置 **-webkit-tap-highlight-color** 的 alpha 值为 0 去除部分机器自带的效果

```
a,button,input,textarea{
    -webkit-tap-highlight-color: rgba(0,0,0,0)
    -webkit-user-modify:read-write-plaintext-only;
}
```

-webkit-user-modify 有个副作用，就是输入法不再能够输入多个字符

另外，有些机型去除不了，如小米 2

对于按钮类还有个办法，不使用 **a** 或者 **input** 标签，直接用 **div** 标签

8、winphone 系统 a、input 标签被点击时产生的半透明灰色背景怎么去掉

```
<meta name="msapplication-tap-highlight" content="no">
```

9、webkit 表单元素的默认外观怎么重置

```
.css{-webkit-appearance:none;}
```

10、webkit 表单输入框 placeholder 的颜色值能改变么

```
input::-webkit-input-placeholder{color:#AAAAAA;}
input:focus::-webkit-input-placeholder{color:#EEEEEE;}
```

11、webkit 表单输入框 placeholder 的文字能换行么

ios 可以，android 不行~

12. 关闭 iOS 键盘首字母自动大写

在 iOS 中，默认情况下键盘是开启首字母大写的功能的，如果启用这个功能，可以这样：

```
<input type="text" autocapitalize="off" />
```

13. 关闭 iOS 输入自动修正

和英文输入默认自动首字母大写那样，IOS 还做了一个功能，默认输入法会开启自动修正输入内容，这样的话，用户经常要操作两次。如果不希望开启此功能，我们可以通过 input 标签属性来关闭掉：

```
<input type="text" autocorrect="off" />
```

14. 禁止文本缩放

当移动设备横竖屏切换时，文本的大小会重新计算，进行相应的缩放，当我们不需要这种情况时，可以选择禁止：

```
html {
    -webkit-text-size-adjust: 100%;
}
```

需要注意的是，PC 端的该属性已经被移除，该属性在移动端要生效，必须设置 `meta viewport`。

15. 移动端如何清除输入框内阴影

在 iOS 上，输入框默认有内部阴影，但无法使用 box-shadow 来清除，如果不需要阴影，可以这样关闭：

```
input,
textarea {
    border: 0; /* 方法 1 */
    -webkit-appearance: none; /* 方法 2 */
}
```

16. 快速回弹滚动

我们先来看看回弹滚动在手机浏览器发展的历史：

- 早期的时候，移动端的浏览器都不支持非 body 元素的滚动条，所以一般都借助 iScroll;
- Android 3.0/iOS 解决了非 body 元素的滚动问题，但滚动条不可见，同时 iOS 上只能通过 2 个手指进行滚动;
- Android 4.0 解决了滚动条不可见及增加了快速回弹滚动效果，不过随后这个特性又被移除;
- iOS 从 5.0 开始解决了滚动条不可见及增加了快速回弹滚动效果

在 iOS 上如果你想让一个元素拥有像 Native 的滚动效果，你可以这样做：

```
.xxx {
    overflow: auto; /* auto | scroll */
    -webkit-overflow-scrolling: touch;
```



```
}
```

PS: iScroll 用过之后感觉不是很好, 有一些诡异的 bug, 这里推荐另外一个 iDangero Swiper, 这个插件集成了滑屏滚动的强大功能(支持 3D), 而且还有回弹滚动的内置滚动条, 官方地址:

[iDangero](#)

17. 移动端禁止选中内容

如果你不想用户可以选中页面中的内容, 那么你可以在 css 中禁掉:

```
.user-select-none {
  -webkit-user-select: none; /* Chrome all / Safari all */
  -moz-user-select: none;    /* Firefox all (移动端不需要) */
  -ms-user-select: none;    /* IE 10+ */
}
```

18. 移动端取消 touch 高亮效果

在做移动端页面时, 会发现所有 a 标签在触发点击时或者所有设置了伪类 :active 的元素, 默认都会在激活状态时, 显示高亮框, 如果不想要这个高亮, 那么你可以通过 css 以下方法来进行全局的禁止:

```
html {
  -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
}
```

但这个方法在三星的机子上无效, 有一种妥协的方法是把页面非真实跳转链接的 a 标签换成其它标签, 可以解决这个问题。

19. 如何禁止保存或拷贝图像 (IOS)

通常当你在手机或者 pad 上长按图像 img, 会弹出选项 存储图像 或者 拷贝图像, 如果你不想让用户这么操作, 那么你可以通过以下方法来禁止:

```
img { -webkit-touch-callout: none; }
```

20. 模拟按钮 hover 效果

移动端触摸按钮的效果, 可明示用户有些事情正要发生, 是一个比较好体验, 但是移动设备中并没有鼠标指针, 使用 css 的 hover 并不能满足我们的需求, 还好国外有个激活 css 的 active 效果, 代码如下,

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no" name="viewport">
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
<meta content="telephone=no" name="format-detection">
```

```

<meta content="email=no" name="format-detection">
<style type="text/css">
a{-webkit-tap-highlight-color: rgba(0,0,0,0);}
.btn-blue{display:block;height:42px;line-height:42px;text-align:center;border-
radius:4px;font-size:18px;color:#FFFFFF;background-color: #4185F3;}
.btn-blue:active{background-color: #357AE8;}
</style>
</head>
<body>

<div class="btn-blue">按钮</div>

<script type="text/javascript">
document.addEventListener("touchstart", function() {}, true)
</script>
</body>
</html>

```

兼容性 ios5+、部分 android 4+、winphone 8

要做到全兼容的办法，可通过绑定 `ontouchstart` 和 `ontouchend` 来控制按钮的类名

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-
scalable=no" name="viewport">
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
<meta content="telephone=no" name="format-detection">
<meta content="email=no" name="format-detection">
<style type="text/css">
a{-webkit-tap-highlight-color: rgba(0,0,0,0);}
.btn-blue{display:block;height:42px;line-height:42px;text-align:center;border-
radius:4px;font-size:18px;color:#FFFFFF;background-color: #4185F3;}
.btn-blue-on{background-color: #357AE8;}
</style>
</head>
<body>

<div class="btn-blue">按钮</div>

<script type="text/javascript">
var btnBlue = document.querySelector(".btn-blue");

```

```

btnBlue.ontouchstart = function(){
    this.className = "btn-blue btn-blue-on"
}
btnBlue.ontouchend = function(){
    this.className = "btn-blue"
}
</script>
</body>
</html>

```

21. 屏幕旋转的事件和样式

事件

window.orientation, 取值: 正负 90 表示横屏模式、0 和 180 表现为竖屏模式;

```

window.onorientationchange = function(){
    switch(window.orientation){
        case -90:
        case 90:
            alert("横屏:" + window.orientation);
        case 0:
        case 180:
            alert("竖屏:" + window.orientation);
            break;
    }
}

```

样式

```

//竖屏时使用的样式
@media all and (orientation:portrait) {
    .css{}
}

//横屏时使用的样式
@media all and (orientation:landscape) {
    .css{}
}

```

22. audio 元素和 video 元素在 ios 和 android 中无法自动播放

应对方案: 触屏即播

```

$('html').one('touchstart',function(){
    audio.play()
})

```

23. 摇一摇功能

HTML5 deviceMotion: 封装了运动传感器数据的事件, 可以获取手机运动状态下的运动加速度等数据。

24. 手机拍照和上传图片

`<input type="file">` 的 accept 属性

```
<!-- 选择照片 -->
<input type=file accept="image/*">
<!-- 选择视频 -->
<input type=file accept="video/*">
```

使用总结:

- ios 有拍照、录像、选取本地图片功能
- 部分 android 只有选取本地图片功能
- winphone 不支持
- input 控件默认外观丑陋

25. 消除 transition 闪屏

```
.css{
  /*设置内嵌的元素在 3D 空间如何呈现: 保留 3D*/
  -webkit-transform-style: preserve-3d;
  /* (设置进行转换的元素的背面在面对用户时是否可见: 隐藏) */
  -webkit-backface-visibility: hidden;
}
```

开启硬件加速

- 解决页面闪白
- 保证动画流畅

```
• .css {
•   -webkit-transform: translate3d(0, 0, 0);
•   -moz-transform: translate3d(0, 0, 0);
•   -ms-transform: translate3d(0, 0, 0);
•   transform: translate3d(0, 0, 0);
• }
```

设计高性能 CSS3 动画的几个要素

- 尽可能地使用合成属性 transform 和 opacity 来设计 CSS3 动画,
- 不使用 position 的 left 和 top 来定位
- 利用 translate3D 开启 GPU 加速

26. android 上去掉语音输入按钮

```
input::-webkit-input-speech-button {display: none}
```